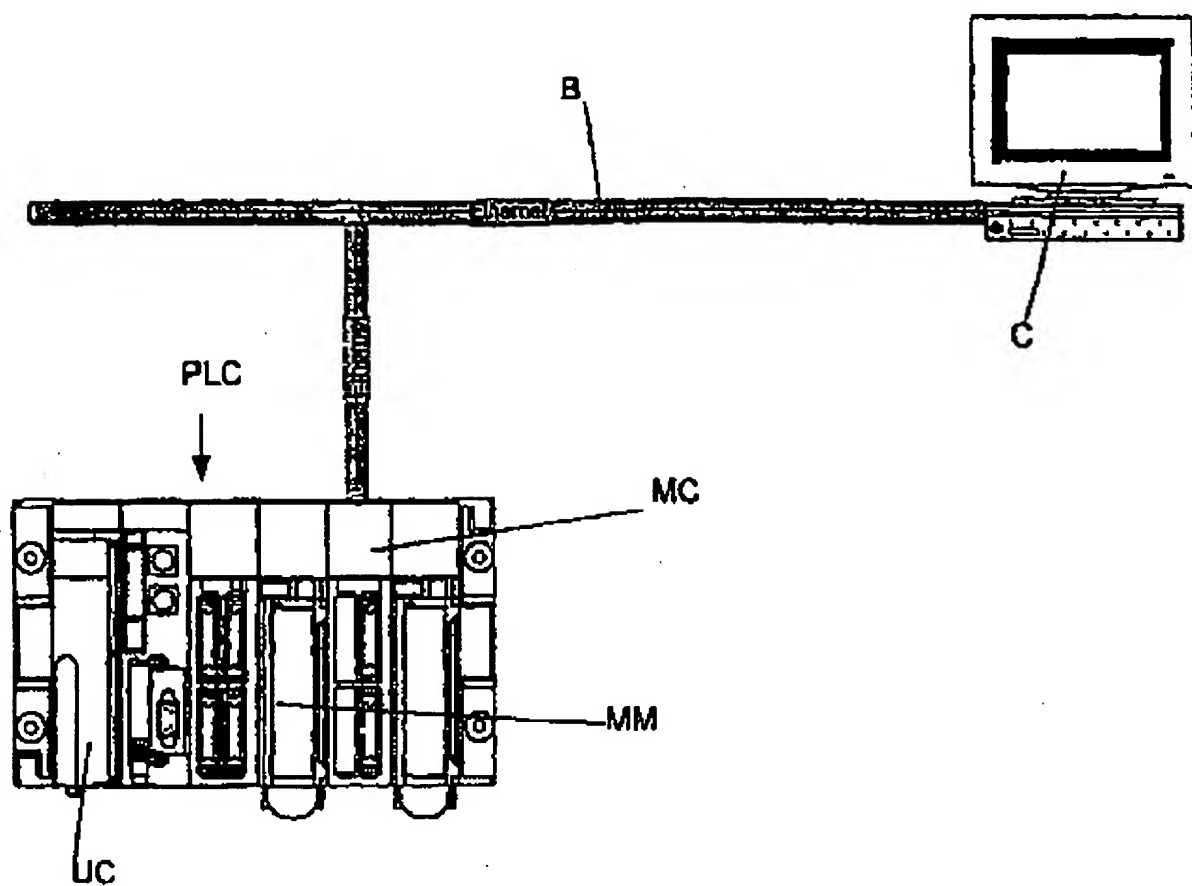


AN: PAT 2002-035430
TI: Programmable controller with communication facilities in client server architecture, comprises intelligent module with user programmable interface which enables connection and communication functions of communication module used remotely
PN: **DE10103533-A1**
PD: 02.08.2001
AB: NOVELTY - The programmable controller (PLC) is designed for use with a private operating system and operates with an automation program. The system has intelligent modules (MM,UC) in which a HTTP server. A communication module (MC) connects over a bus with a remote system (C). The intelligent module is provided with a user programmable interface which enables the connection and (TCP) communication functions of the TCP/IP-type communication module to be used remotely.; USE - For use in automation systems. ADVANTAGE - Allows client function or sever function to be run or to be a member of a multi-cast group while using the TCP/IP-service of the communication module of the configuration of the programmable controller. DESCRIPTION OF DRAWING(S) - The drawing shows a block diagram. Bus B Client system C Intelligent modules MM,UC Communication module MC Program regulator PLC
PA: (SCHN-) SCHNEIDER AUTOMATION;
(SCHN-) SCHNEIDER AUTOMATION SA;
(SCHN-) SCHNEIDER ELECTRIC IND SA;
IN: BRAULT G;
FA: **DE10103533-A1** 02.08.2001; US6721607-B2 13.04.2004;
FR2804218-A1 27.07.2001; US2001014833-A1 16.08.2001;
CO: DE; FR; US;
IC: G05B-011/01; G05B-019/04; G05B-019/05; G06F-013/10;
G08C-019/00; H04L-012/28;
MC: T01-F06; T01-N02A1; T01-N02A2C; T06-A04B1; W05-D02; W05-D03;
DC: T01; T06; W05;
FN: 2002035430.gif
PR: FR0001015 26.01.2000;
FP: 27.07.2001
UP: 16.04.2004



BEST AVAILABLE COPY



⑬ BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

Offenlegungsschrift DE 101 03 533 A 1

⑤ Int. Cl.⁷:
G 05 B 19/04
G 08 C 19/00

⑲ Aktenzeichen: 101 03 533.0
⑳ Anmeldetag: 26. 1. 2001
㉑ Offenlegungstag: 2. 8. 2001

DE 101 03 533 A 1

③① Unionspriorität:
00 01015 26. 01. 2000 FR

⑦① Anmelder:
Schneider Automation, Valbonne, FR

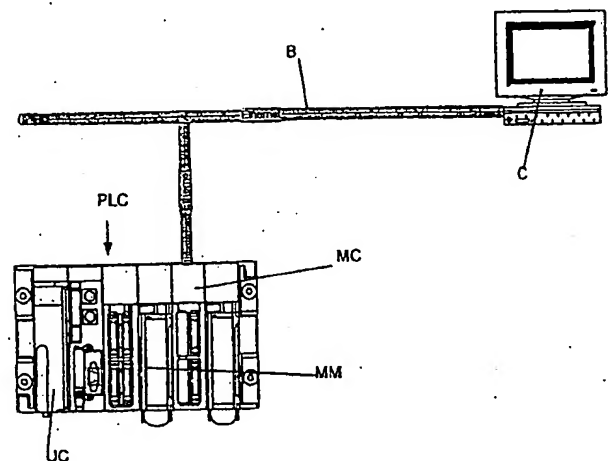
⑦④ Vertreter:
Manitz, Finsterwald & Partner GbR, 80538 München

⑦② Erfinder:
Brault, Gilbert, Vence, FR

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

⑤④ Programmregler mit Kommunikationen in einer Client/Server-Architektur

⑤⑦ Die Erfindung betrifft einen Programmregler (PLC), der mit Kommunikationsfunktionen in einer Client/Server-Architektur und mit einem privaten Betriebssystem versehen ist, das nicht das TCP/IP-Protokoll unterstützt, sich jedoch für Automatisierungsprogramme eignet, und der einerseits mit mindestens einem intelligenten Modul (MM, UC), in dem ein ATTP-Server installiert ist; und andererseits mit einem Kommunikationsmodul (MC), das über einen Bus mit entfernten Einrichtungen (C) kombiniert werden kann, ausgestattet ist, dadurch gekennzeichnet, dass das intelligente Modul (MM, UC) mit einer Anwendungsprogrammierschnittstelle (TCP API) versehen ist, die es ermöglicht, Verbindungs- und (TCP-) Kommunikationsfunktionen des TCP/IP-artigen Kommunikationsmoduls (MC) entfernt zu verwenden.



DE 101 03 533 A 1

Beschreibung

Die vorliegende Erfindung betrifft einen Programmregler, der mit einem privaten Betriebssystem versehen ist, welches nicht das TCP/IP-Protokoll unterstützt, sich jedoch für Automatisierungsprogramme eignet, und der einerseits mit mindestens einem intelligenten Modul, in dem ein HTTP-Server installiert ist, und andererseits mit einem Kommunikationsmodul, das über einen Bus mit entfernten Einrichtungen kombiniert werden kann, ausgestattet ist.

Ein modularer Programmregler besteht aus einer programmierbaren Prozessoreinheit und intelligenten Modulen, die Automatisierungsfunktionen übernehmen (z. B. Wägung, Achsensteuerung, Regelung, usw.) und an die Prozessoreinheit über einen Kommunikationsbus oder Netzwerke mit privaten Protokollen angeschlossen sind. Die Hardware- und Softwaretechnologie (Echtzeitkern, Betriebssystem und Kommunikationsprotokoll) des Automaten ist meistens "proprietär", d. h. spezifisch für den Automatenhersteller. Die programmierbare Prozessoreinheit enthält die in der Automatenprogrammiersprache geschriebene Anwendung (z. B. in einer unter der Bezeichnung PL7 bekannten Sprache für die Automaten von Schneider Automation). Ein intelligentes Modul ist ein Mikrocomputer, der mit einer Prozessoreinheit, Festwert- und Direktzugriffsspeicher sowie privaten Ein-/Ausgaben versehen ist.

Der Automat kann außerdem mit einem Kommunikationsmodul ausgestattet sein, das den Anschluss an ein TCP/IP-artiges Netzwerk und AN/AUS oder analogen Ein-/Ausgabemodulen ermöglicht.

Ein TCP/IP-Kommunikationsmodul implementiert Serverfunktionen (z. B. die HTTP-Serverfunktion), die es einem TCP/IP-Client ermöglichen, sich über ein Netzwerk mit TCP/IP-Protokoll an den Automaten anzuschließen. Es sei darauf hingewiesen, dass ein TCP/IP-Client eine Einrichtung ist, welche die Initiative zur Eröffnung einer Verbindung auf einem TCP/IP-Server hat. Es ist ebenfalls möglich, dass das TCP/IP-Kommunikationsmodul eine Clientfunktion auf TCP/IP implementiert, die den Anschluss an eine TCP/IP-Servereinrichtung ermöglicht.

Das TCP/IP-Kommunikationsmodul kann als Brücke dienen, um die Datenübertragungsblöcke eines Netzwerks mit privatem Protokoll bis zu einem Netzwerk mit TCP/IP-artigem Protokoll zu transportieren. Damit das Kommunikationsmodul Dienste nach Art eines FTP-Übertragungsservers ("File Transfer Protocol"), eines HTTP-Servers ("Hypertext Transfer Protocol") oder eines SNMP-Agenten ("Simple Network Management Protocol") anbieten kann, stellt die Zentraleinheit des Programmreglers Datenzonen frei, welche die verschiedenen Dienste den entfernten Clientstationen des TCP/IP-Netzwerks präsentieren. Dienste nach Art eines "Mail"-Clients, eines FTP-Clients, eines SMTP-Clients ("Simple Network Time Protocol") oder eines SAP-Clients (Client-/Serverunternehmensverwaltungssoftware, insbesondere mit Produktionsverwaltungsfunktionen) können es dem Anwendungsprogramm des Automaten erlauben, den Speicher mit Informationen zu versorgen, die von entfernten Servern, die diese Protokolle implementieren, stammen. Die Client- oder Serverfunktionen werden im TCP/IP-Kommunikationsmodul vorgegeben und kodiert. Damit Dienste der intelligenten Module vom Netzwerk aus zugänglich sind, muss eine anpassungsfähige Anwendung in der Zentraleinheit des Automaten implementiert werden.

Ziel der Erfindung ist es, einem intelligenten Modul die Möglichkeit zu geben, eine Clientfunktion oder eine Serverfunktion ablaufen zu lassen, oder Mitglied einer Multicastgruppe zu sein, indem die TCP/IP-Dienste eines Kommunikationsmoduls der Konfiguration des Programmreglers ver-

wendet werden. Demnach kann das intelligente Modul von einem Mitglied des TCP/IP-Netzwerks verwendet werden, als ob es damit physikalisch kombiniert wäre.

Der erfindungsgemäße Automat ist dadurch gekennzeichnet, dass das intelligente Modul mit einer Anwendungsschnittstelle versehen ist, die es ermöglicht, entfernt die Anschluss- und Kommunikationsfunktionen des TCP/IP-artigen Kommunikationsmoduls zu verwenden.

Die Erfindung wird nun unter Bezugnahme auf eine Ausgestaltung, die als Beispiel angegeben wird und in den beigefügten Zeichnungen dargestellt ist, näher beschrieben. Es zeigen:

Fig. 1 ein Diagramm der Hardware-Architektur eines erfindungsgemäßen, modularen Programmreglers.

Fig. 2 ein Blockdiagramm, das die Kommunikationsfunktionen eines Kommunikationsmoduls und eines intelligenten Moduls veranschaulicht.

Fig. 3 veranschaulicht die verschiedenen Anschluss- und Kommunikationsfunktionen zwischen einem Client und einem Server.

Die in Fig. 1 dargestellte Architektur umfasst einen mit PLC bezeichneten Automaten, der mit einer Zentraleinheit UC, intelligenten Modulen MM mit Automatisierungsfunktion (Wägung, Achsensteuerung, ...) und einem Kommunikationsmodul MC mit Serverfunktion ausgestattet ist. Diese Module und Zentraleinheit kommunizieren über einen privaten Kommunikationsbus BP, der beispielsweise ein gestellmontierter Bus ist. Dieser Serverautomat PLC kann mit anderen Programmreglern kombiniert werden.

Die Zentraleinheit UC des Programmreglers enthält ein Anwendungsprogramm, das in einer üblichen Automatisierungsbefehlssprache ("Kontaktsprache", usw.) geschrieben ist. Es verwaltet die Ein-/Ausgaben durch die Verwendung von Automatenvariablen.

Das mit einem Zentralspeicher und einem Mikroprozessor ausgerüstete Kommunikationsmodul MC ist mit einem Echtzeitkern versehen, der durch eine TCP/IP-Netzwerkschicht ("stack TCP/IP") erweitert ist, die einen Hardwareadapter steuert, der es ihr ermöglicht, über ein Netzwerk B mit TCP/IP-Protokoll mit einem oder mehreren Clientstationen C (Computern) zu kommunizieren.

Die Software-Architektur des Kommunikationsmoduls MC besteht aus:

- Einem Server-Delegationsdienst, der es einer Automatisierungsfunktion eines intelligenten Moduls mm erlaubt, dem Modul MC die Eröffnung und Verwaltung einer Serververbindung auf dem TCP/IP-artigen Netzwerk B zu delegieren.
- Einem Client-Delegationsdienst, der es einer Automatisierungsfunktion eines intelligenten Moduls mm erlaubt, dem Modul MC die Eröffnung und Verwaltung einer Clientverbindung auf dem TCP/IP-artigen Netzwerk B zu delegieren.
- Einem Multicast-Delegationsdienst, der es einer Automatisierungsfunktion eines intelligenten Moduls mm erlaubt, dem Modul MC die Eröffnung und Verwaltung einer Multicastverbindung auf dem TCP/IP-artigen Netzwerk B zu delegieren.
- Einem Kodier-/Dekodier- und Verteilerdienst, der es einerseits ermöglicht, von den vorhergehenden Diensten stammende Abfragen nach einem der von dem privaten Kommunikationsbus BP unterstützten Protokolle zu kodieren, sie zu verteilen, um sie den intelligenten Modulen zuzuführen, und andererseits, von den intelligenten Modulen MM stammende Abfragen zu dekodieren, und sie auf einen der oben definierten Delegationsdienste zu verteilen.

Die Software-Architektur des intelligenten Moduls mm umfasst eine Anwendungsprogrammierschnittstelle (API) TCP API, die es ermöglicht, die delegierten Dienste des Kommunikationsmoduls MC entfernt zu verwenden. Die Schnittstelle TCP API kann deshalb von dem Programmierer verwendet werden, der die Automatisierungsfunktion des intelligenten Moduls mm nach drei verschiedenen Nutzungen entwickelt (oder ergänzt); um folgendes zu erzielen:

- Eine Serverfunktion: das ist die Schnittstelle "Server TCP API"
- Eine Clientfunktion: das ist die Schnittstelle "Client TCP API"
- Eine Funktion, um an einer Multicastgruppe teilzunehmen: "Multicast TCP API".

Mit Bezug auf Fig. 3 besteht die Schnittstelle Server TCP API aus den Funktionen, die es der Serveranwendung des intelligenten Moduls mm des Automaten PLC erlauben, mit der entfernten Clientstation C zu kommunizieren, also:

- "Socket", die einen Abhörsocket für die Vorbereitung einer Verbindung erstellt. Ein Socket ist eine Datenstruktur, die Informationen materialisiert, die es dem Programm der Automatisierungsfunktion ermöglichen, auf die Netzwerkschichten zuzugreifen.
- "Bind", die eine Anschlussnummer mit einem Socket verknüpft.
- "Listen", die einen Server-artigen Socket auf das Abhören von Clientverbindungen einstellt.
- "Accept", die einen Clientsocket erstellt, wenn ein Client sich entfernt verbindet.
- "Shutdown", "Close", die das Schließen der Verbindung ausführen und die verwendeten Ressourcen zurückgeben.

Nach der Verbindung werden durch die folgenden Funktionen zwei Daten-(Zeichen-)ströme zwischen Client und Server hergestellt:

- "Recv", der den Empfang der Zeichen des TCP/IP-Clientdatenstroms ermöglicht,
- "Send", der Zeichen an den TCP/IP-Clientstrom schickt.

Mit Bezug auf Fig. 3 besteht die Schnittstelle Client TCP API aus den Funktionen, die es der Clientautomatisierungsfunktion des intelligenten Moduls mm des Automaten PLC erlauben, mit der entfernten Serverstation C zu kommunizieren, also:

- "Socket", die einen Socket erstellt. Ein Socket ist eine Datenstruktur, die Informationen materialisiert, die es dem Programm der Automatisierungsfunktion ermöglichen, auf die Netzwerkschichten zuzugreifen.
- "Connect", die den Socket auf dem entfernten Server verbindet.
- "Shutdown", "Close", die das Schließen der Verbindung ausführen und die verwendeten Ressourcen zurückgeben.

Nach der Verbindung werden durch die folgenden Funktionen zwei Daten-(Zeichen-)ströme zwischen Client und Server hergestellt:

- "Recv", der den Empfang der Zeichen des TCP/IP-Clientdatenstroms ermöglicht.

- "Send", der Zeichen an den TCP/IP-Clientstrom schickt.

Die Automatisierungsfunktion des intelligenten Moduls MM für die Teilnahme an einer Multicast-Kommunikationsgruppe verwendet eine Anwendungsprogrammierschnittstelle Multicast TCP API, die es ermöglicht, die folgenden Funktionen als Abfragen zu kodieren/dekodieren:

- Socket Erstellung eines Sockets für die Vorbereitung einer Verbindung
- Join Verbindung des Sockets bei einer Multicastgruppe
- Recv Empfang von Zeichen des Datenstroms der Multicastgruppe
- Send Sendung von Zeichen des Datenstroms der Multicastgruppe
- Leave Unterbrechung des Sockets der Multicastgruppe
- Close Dienstfunktionen, welche die Rückgabe der verwendeten Ressourcen ermöglichen

Die Funktionen der oben genannten Schnittstelle TCP API (Server, Client, Multicast) verhalten sich wie die gleichnamigen Funktionen der Bibliothek libc des c-Kompilers von Gnu. Die Funktionen von TCP API haben jedoch zusätzliche Parameter (Argumente) im Verhältnis zu den entsprechenden libc-Funktionen. Diese Argumente ermöglichen die Kodierung der Informationen, um sie auf dem Kommunikationsbus zu transportieren und sie mit den delegierten Diensten des Moduls MC zu interpretieren, und die von der Schnittstelle TCP API stammenden Abfragen auf dem privaten Kommunikationsbus BP zu verteilen (Adressierung der Module auf dem privaten Kommunikationsbus, ...).

Zu diesen Argumenten gehört auch ein "Statusblock", der den Asynchronismus zwischen der Automatisierungsfunktion des intelligenten Moduls mm und den delegierten Diensten des Moduls MC ermöglicht. Dazu schickt der erste Aufruf einer Funktion eine Abfrage an den delegierten Dienst des Kommunikationsmoduls MC und initialisiert den Statusblock. Die folgenden Aufrufe mit demselben Statusblock ermöglichen es, die Rückgabeparameter der Funktion einzulesen: wenn die Funktion im Modul MC beendet ist, schickt sie eine Nachricht an das Modul MM. Die Automatisierungsfunktion des intelligenten Moduls MM kann damit Aktionen, bei denen z. B. die Schnittstellen verwendet werden, die das Modul mit dem von ihm gesteuerten Prozess (Prozessschnittstelle) verbinden, zwischen den oben beschriebenen aufeinander folgenden Aufrufen ausführen: dies nennt man Asynchronismus.

Die Automatisierungsfunktion des Moduls mm wird über eine Schnittstelle mit einer Server-, Client oder Multicastgruppenfunktion verbunden, indem für die Betriebsarten (Ablauf der Vorgänge, die implementiert werden müssen, um die entsprechenden Informationen für die gesteuerten Prozesse zu liefern oder auf sie einzuwirken) und die Zuordnung der Anschlussnummern (im Falle eines Servers) oder der entfernten Adressen für Clientfunktionen mit dem Automatenanwendungsprogramm Konventionen übernommen werden, die zum Bereich der Systemanalyse gehören.

Die Schnittstelle TCP API kann in der Zentraleinheit UC des Programmreglers, der ein spezifisches intelligentes Modul ist, implementiert werden, wobei die unterstützte Automatisierungsfunktion Automatenanwendungsprogramm heißt.

Das Anwendungsprogramm des Programmreglers verwendet eine Ergänzung des Systembefehlsvorrats in der Au-

tomatisierungssprache (für die Automaten von Schneider Automation unter der Bezeichnung PL7 bekannt), die den delegierten Server-, Client- oder Multicastdiensten entspricht, um die Server-, Client- oder Multicastfunktionen auszuführen.

Eine derartige Anwendung erlaubt es dem Automaten, der Client oder Server einer beliebigen, entfernten Server- oder Clientanwendung zu sein.

Ein im intelligenten Modul mm implementierter HTTP-Server kann z. B. die Konfiguration dieses Moduls einer mit einem Webnavigator ausgestatteten Clientstation über direkt erzeugte HTML-Seiten freistellen. Somit ist es möglich, dieses intelligente Modul MM online oder nicht online zu konfigurieren.

Nachstehend wird die Arbeitsweise für einen Automaten beschrieben, der in seiner Konfiguration ein TCP/IP unterstützendes Kommunikationsmodul MC umfasst.

Beim Start des Automaten startet das intelligente Modul MM seinen HTTP-Server. Zur Zuordnung der Anschlüsse des jeweiligen Servers gehört z. B. die geografische Nummer des intelligenten Moduls MM, um eine einzigartige und einfache Identifizierung für die Person zu ermöglichen, welche die Anwendung des Programmreglers programmiert, aktualisiert oder verwendet.

Gegebenenfalls können Java-Applets die Funktionen der dynamischen Auffrischung übernehmen.

Ergänzend zum Online-Modus kann ein virtuelles Modul während der Konfiguration des Automaten installiert werden, das auf der Programmierstation einen Simulationsserver startet, wodurch Funktionen wie die des im Modul integrierten Servers erbracht werden.

Es ist selbstverständlich, dass man sich, ohne den Rahmen der Erfindung zu verlassen, Variationen und Detailweiterentwicklungen ausdenken und auch den Gebrauch gleichwertiger Mittel in Betracht ziehen kann.

Patentansprüche

1. Programmregler (PLC), versehen mit einem privaten Betriebssystem, welches nicht das TCP/IP-Protokoll unterstützt, sich jedoch für Automatisierungsprogramme eignet, und einerseits mit einem intelligenten Modul (MM, UC), in dem ein HTTP-Server installiert ist, und andererseits mit einem Kommunikationsmodul (MC), das über einen Bus mit entfernten Einrichtungen (C) kombiniert werden kann, ausgestattet ist, **dadurch gekennzeichnet**, dass das intelligente Modul (MM, UC) mit einer Anwendungsprogrammierschnittstelle (TCP API) versehen ist, die es ermöglicht, entfernt die Verbindungs- und (TCP-) Kommunikationsfunktionen des TCP/IP-artigen Kommunikationsmoduls (MC) zu verwenden.

2. Regler nach Anspruch 1, dadurch gekennzeichnet, dass das Kommunikationsmodul (MC) einen Server-Delegationsdienst, der es einer Anwendung eines Moduls (MM) erlaubt, dem Modul (MC) die Eröffnung und Verwaltung einer Serververbindung auf dem TCP/IP-Netzwerk zu delegieren, einen Client-Delegationsdienst, der es einer Anwendung eines Moduls (MM) erlaubt, dem Modul (MC) die Eröffnung und Verwaltung einer Clientverbindung auf dem TCP/IP-Netzwerk zu delegieren, einen Multicast-Delegationsdienst, der es einer Anwendung eines Moduls (MM) erlaubt, dem Modul (MC) die Eröffnung und Verwaltung einer Multicastverbindung auf dem TCP/IP-Netzwerk zu delegieren, und einen Transkodier- und Routingdienst, der es ermöglicht, die von den vorhergehenden Diensten stammenden Abfragen nach einem der von

dem privaten Kommunikationsbus unterstützten Protokollen zu kodieren, um sie den intelligenten Modulen zuzuführen, und die von den intelligenten Modulen stammenden Abfragen zu dekodieren und sie auf die Delegationsdienste zu verteilen, umfasst.

3. Regler nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass die Anwendungsprogrammierschnittstelle (TCP API) Funktionen umfasst, die sich wie gleichnamige Funktionen der Bibliothek libc des c-Kompilers von gnu verhalten, jedoch zusätzliche Argumente haben, um die Informationen derart zu kodieren, dass sie auf dem privaten Kommunikationsbus (BP) transportiert werden können und dass sie von den delegierten Diensten des Kommunikationsmoduls (MC) und der Verteilung der von der sogenannten Schnittstelle (TCP API) auf dem privaten Kommunikationsbus (BP) stammenden Nachrichten interpretiert werden können.

4. Regler nach Anspruch 1, 2 oder 3, dadurch gekennzeichnet, dass zur Sicherstellung des Asynchronismus zwischen der Anwendung des Kommunikationsmoduls (MM) und den delegierten Diensten des Kommunikationsmoduls (MC) der erste Aufruf einer Funktion eine Anfrage an den delegierten Dienst des Kommunikationsmoduls (MC) schickt und einen als Parameter weitergegebenen Statusblock initialisiert, und dass die folgenden Aufrufe mit demselben Statusblock das Einlesen der Rückgabeparameter der Funktion ermöglichen, wobei eine Nachricht an das intelligente Modul (MM) geschickt wird, wenn die Funktion im Modul (MC) beendet ist.

Hierzu 2 Seite(n) Zeichnungen

BEST AVAILABLE COPY

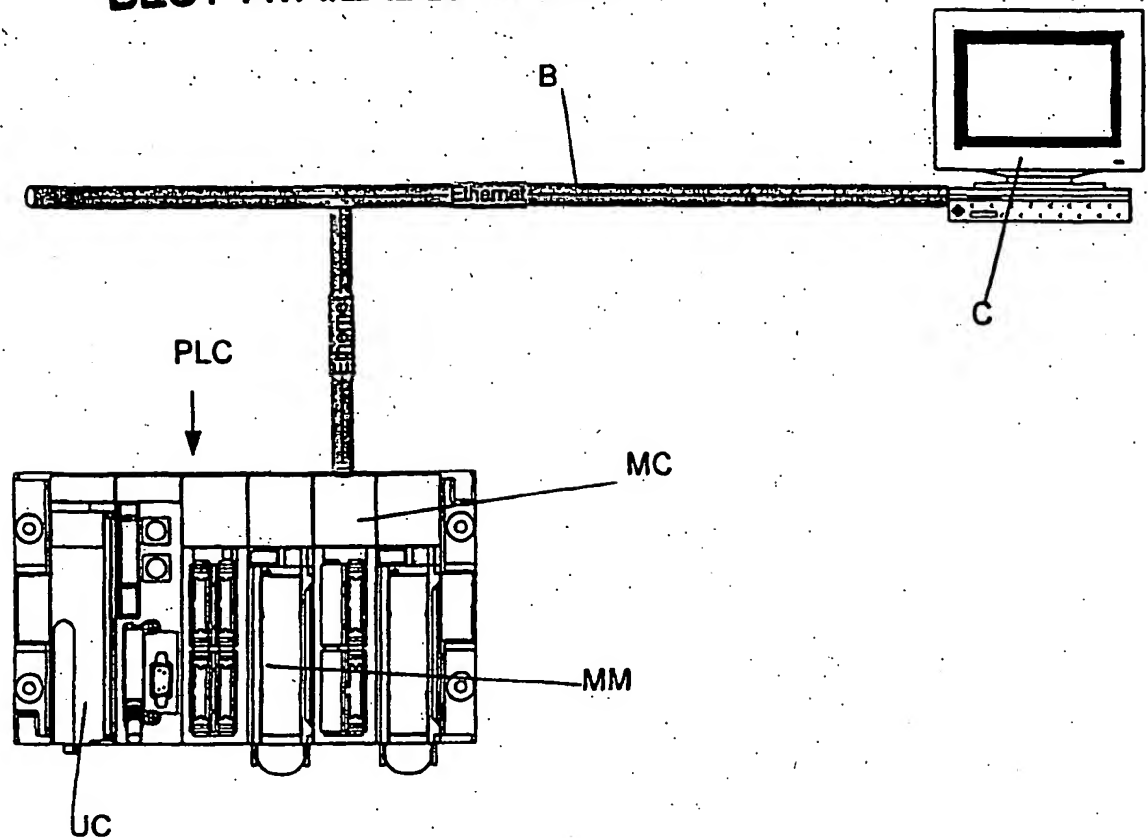


Fig. 1

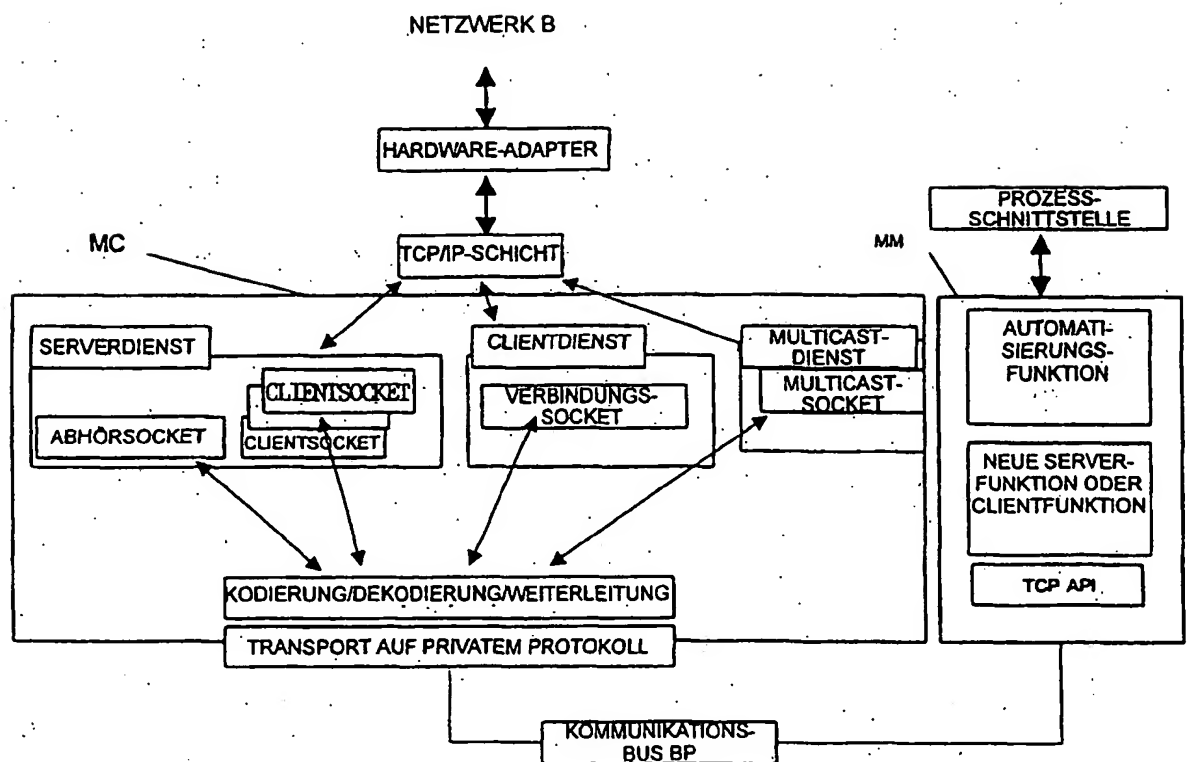


Fig. 2

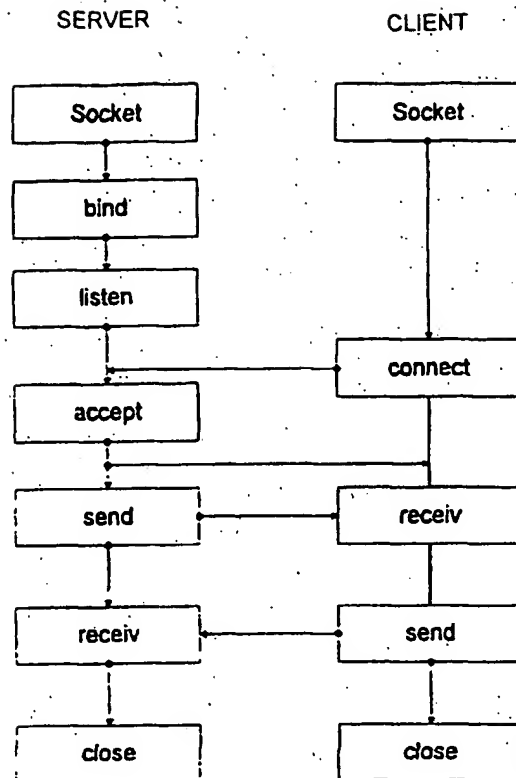


Fig. 3